Commodore

comments and bulletins
concerning your
COMMODORE PET

The Transactor

BULLETIN # 10

PET[tm] is a registered trademark of Commodore Inc.

March 31, 1979

Included in this months Transactor is a list of the 6502 instruction set; mnemonics and what they stand for followed by their respective machine language coding and addressing modes.

When operating in machine language, PET is at top efficiency. Machine code programs can execute at speeds 10 to 1000 times that of the equivalent BASIC implementation. Also, depending on the operation, they may consume as much as 10 times less memory. The reason BASIC is so "slow" is that BASIC must first be interpreted into machine code such that the Microprocessor can handle it. In fact, about 90% of the total execution time is spent interpreting while only about 10% of the time is spent on the actual operation. In machine code programs the BASIC interpreter is bypassed hence the greatly increased speed of processing. This speed is realized most in programs where a lot of tests or comparisons are made. MicroChess[c] is a prime example. At level 8 (playing at it's best) the machine can still spend as much as 10 minutes on a move in some situations. Imagine a chess program coded in BASIC!

A brief explanation of machine language would be highly impracticle because of the variations and possibilities of the concepts. Undoubtedly a lot of imprtant information would be overlooked. However I have here an exerpt from PET User Notes, Issue #5, written by Jim Butterfield:

"A Little Exercise in PET Machine Language
Jim Butterfield Toronto, Canada

Clear the PET completely (NEW:CLR) and enter the following three lines of BASIC...Do not insert extra spaces!

```
100 SYS(1050)
110 GOTO100
120 XXXXXXXXXXXXXXXXXXXXXXXXX
```

The last line should not be less than 15 X's, and preferably a few more, say 25. You may list this program but do not try to run yet.

Now you have some POKEing to do, and unfortunately you can't have a program help you. First, make sure that the above lines are OK by ?PEEK(1050); this should return an 88 (X character). Now starting with POKE1050 ,32...input the following values:

```
Starting at 1050:  32 228 255 208   1  96 162   0 157   0
Starting at 1060: 128 157   0 129 157   0 130 157   0 131
Starting at 1070: 202 208 241  96
```

Double-check the above values by listing them with:

```
FOR J=1050 TO 1070 STEP 10:FOR K=0 TO 9:?PEEK(J+K);:
NEXTK:?:NEXTJ
```

It is vital that these numbers be correct - one mistake and your system will crash. Behind the 96 you should see some leftover X's (88's).

Now type RUN. Try tapping a few keys and note how the screen changes. Stop the program with the STOP key.

What's it all about? We've written a program in machine language, the fundamental 6502 language of the PET. In working with the inner fabric of the machine we find we get: (i) compactness - we've fitted a whole program within one BASIC line: (ii) speed - no BASIC program could fill the screen that fast. We lose, however, in the need for preciseness; one mistake and the system crashes, and you have to switch off and on again. We also lose flexibility - adding an instruction isn't easy.

For those who would like to try tracking the machine language program above, a few brief notes. 32-228-255 calls the PET subroutine to get a character (something like BASIC GET). 208-1--96 exits if no character is seen (like IF X$=" " THEN RETURN). Now we're ready to zip through the screen with the character we found. We set up for repetition with 162-0 which loads an internal (X) register for 256 repeats; much later we invoke the repetition with 202-208-241, and after the 256th time we return (96). Within the repetition itself, we set the four quarters of the screen with four 157-0-xx instructions."

Those interested in getting seriously involved in machine language should consider first the MOS KIM-1 Microcomputer Module and:

The First Book of KIM By: Jim Butterfield, Stan Ockers and Eric Rehnke
Publisher: Hayden

The book is mostly machine language programs written for the KIM. Programming them into PET would be most difficult even with the Machine Language Monitor. KIM has numerous subroutines in ROM that aren't like PET's.

Other suggested reading:

Programming a Microcomputer By: Caxton C. Foster
Publisher: Addison-Wesley

MOS 6500 Programming Manual By: Commodore/MOS Technology

6502 User Notes By: Eric Rehnke
P.O. Box 33093
North Royalton, Ohio
44133
$12.00/Yr.?

All four publications are excellent, but for beginners I suggest the last three in order (if using PET with the M.L.M.).

Besides the program outlined in J. Butterfields article are four more that also operate in machine language. The first three read from the DATA statements data which has already been converted into decimal. This data represents the machine language instructions which (in these three particular programs) are POKEd into the second cassette buffer (826 to 1017). Since they are in decimal, a conversion program has been included so you may convert back to HEX and compare them to the table. However, not all of these will necessarily be instructions (as you will see when you find one that matches a "Future Expansion" code). Some may be addresses of direct data depending on the preceeding instruction. Addresses will appear as low order first, high order second. For example:

JSR 00 05

....will jump to the subroutine starting at location 0500.

The DATA statements in Life contain the actual hexadecimal representations of the instructions and addresses. They are read by the program (line 110), tested for validity (lines 120 and 150), converted to decimal (lines 130 and 140) and POKEd into memory (line 160) starting at decimal location 6400 (HEX 1900). SYS 6400 executes the program.

View By Jim Butterfield
from an idea by Brad Templeton

```
10 PRINT"SYS826 TURNS PAGEVIEW ON AND/OR OFF
20 PRINT:PRINT"SELECT PAGES WITH 'POKE849,X'
30 PRINT TAB(10)"TRY X=0,2,4,31,232
40 FOR J=826 TO 858:READ X:POKE J,X:NEXT: END
50 DATA 120,173,25,2,73,200,141,25,2
60 DATA 173,26,2,73,229,141,26,2,88,96
70 DATA 162,0,189,0,0,157,0,128,202
80 DATA 208,247,76,133,230
```

Non-Stop By Jim Butterfield

```
0 REM**MACHINE LANGUAGE STOP KEY DISABLE**
1 GOSUB 63520:END
63520 DATA 120,169,96,141,25,2,169,3
63521 DATA 141,26,2,88,96,0,0,0
63522 DATA 120,169,133,141,25,2,169,230
63523 DATA 141,26,2,88,96,0,0,0
63524 DATA 32,234,255,169,255,141,9,2
63525 DATA 76,136,230
63526 FOR L=832 TO 874:READ K:POKE L,K:NEXT
63527 RETURN
```

Auto-Repeat

By The Software Shoppe
From 'The Paper' Volume 1, Issue #10

```
5 REM**MACHINE LANGUAGE AUTO-REPEAT**
10 DATA 120,56,169,233,237,26,2,141
20 DATA 26,2,88,96,173,35,2,201,255
30 DATA 208,12,169,0,141,119,3,169
40 DATA 90,141,120,3,208,25,238,119
50 DATA 3,173,120,3,205,119,3,176,14
60 DATA 169,6,141,120,3,162,255,142
70 DATA 3,2,232,142,119,3,76,133,230
80 FOR I=889 TO 947
90 READ J
100 POKE I,J
110 NEXT I
120 PRINT"SYS889 WILL ENABLE AND DISABLE
130 PRINT"THE AUTO REPEAT FUNCTION
140 END
200 REM**TRY THIS AUTO-REPEAT IN BASIC**
210 GET A$
220 PRINT A$;
230 POKE 515,255
240 GOTO 200
```

Life

By Mark Taylor

```
100 READL
110 READA$:C=LEN(A$):IFA$="*"THENEND
120 IFC<10RC>2THEN200
130 A=ASC(A$)-48:B=ASC(RIGHT$(A$,1))-48
140 N=B+7*(B>9)-(C=2)*(16*(A+7*(A>9)))
150 IFN<0ORN>255THEN200
160 POKEL,N:L=L+1:GOTO110
200 PRINT"BYTE"L"=["A$"] ???":END
300 DATA6400
310 DATA20,30,19,20,8A,19,20,E6,19,20,00,1A,A9,34,8D,11,E8
320 DATA20,70,19,A9,3C,8D,11,E8,A9,FF,CD,12,E8,F0,E6,4C,8B,C3,AA,68,28,4C,8B,C3
330 DATA EA,EA,EA,EA,EA,EA,EA,A2,19,BD,3A,19,95,1F,CA,D0,F8,60,00,80,00,15,00
340 DATA80,00,1B,00,1B,D7,28,01,FE,D8,D6,29,27,00,E8,83,00,15,00,00
350 DATAEA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA,EA
360 DATAEA,EA,EA,EA,EA,20,A6,19,B1,26,D0,06,A9,20,91,20,D0,04,A9,51,91,20,20
370 DATA BD,19,F0,ED,20,A6,19,60,20,A6,19,B1,20,C9,51,F0,06,A9,00,91,26,F0
380 DATA04,A9,01,91,26,20,BD,19,F0,EB,20,A6,19,60,A9,00,AA,A8,85,20,85,26,85
390 DATA39,A5,25,85,21,A5,29,85,27,A5,36,85,3A,60,E6,26,E6,20,E6,39,E8,E4
400 DATA33,F0,0C,E0,00,D0,0E,E6,27,E6,21,E6,3A,D0,06,A5,34,C5,21,F0,03,A9,00
410 DATA 60,A9,01,60,EA,EA,EA,EA,EA,EA,20,A6,19,B1,26,D0,06,A9,20,91,39,D0
420 DATA04,A9,51,91,39,20,BD,19,F0,ED,20,A6,19,60,20,A6,19,20,2F,1A,B1,39,C9
430 DATA51,F0,0C,A5,32,C9,03,D0,14,A9,01,91,26,D0,0E,A5,32,C9,03,F0,08,C9,02
440 DATAF0,04,A9,00,91,26,20,BD,19,F0,D8,20,A6,19,60,98,48,8A,48,A0,00,84,32
450 DATAA2,08,B5,29,10,15,49,FF,85,37,38,A5,39,E5,37,85,22,A5,3A,85,23,B0,11
460 DATAC6,23,D0,0D,18,65,39,85,22,A5,3A,85,23,90,02,E6,23,B1,22,C9,51,D0,02
470 DATAE6,32,CA,D0,CF,68,AA,68,A8,68,*
```

```
100 DIM A$(3)
110 FOR K=0 TO 39:PRINT "•";:NEXT:A=0:B=0:C=0:L=0:M=0:N=0:X=0
120 PRINT:INPUT"HEX$crcrcrcrcrcrcrcr";HEX$:L=LEN(HEX$)
130 IF L<>4 AND L<>2 THEN PRINT" 2 OR 4 HEX DIGITS ONLY":GOTO 110
140 IF L=4 THEN C=2:A$(2)=LEFT$(HEX$,2)
150 A$(1)=RIGHT$(HEX$,2):C=2
160 X=X+1
170 A=ASC(A$(X))-48
180 IF A<0 OR A>22 THEN PRINT "!HEX DIGITS ONLY":GOTO 110
190 B=ASC(RIGHT$(A$(X),1))-48
200 IF B<0 OR B>22 THEN PRINT  "!HEX DIGITS ONLY":GOTO 110
210 N=B+7*(B>9)-(C=2)*(16*(A+7*(A>9)))
220 IF L=0 THEN M=M+N:PRINT"DEC ADDRESS = "M:GOTO 110
230 IF L=4 THEN L=0:M=N*256:GOTO 160
240 PRINT"IN DEC =crcrcrcrcr"N:GOTO 110


100 REM DECIMAL TO BINARY TO HEXADECIMAL CONVERTER FROM TRANSACTOR ISSUE NO. 10
110 REM WRITTEN BY KARL J. HILDON ABSOLUTELY NO COPYRIGHTS
120 DIM A$(16)
130 INPUT"DEC VALUE      ";D$
140 A=VAL(D$)
150 IF A-INT(A)<>0 THEN PRINT:PRINT"!INTEGERS ONLY  ":PRINT:GOTO130
160 IF 0>65535 THEN PRINT:PRINT"!65535 MAXIMUM  ":PRINT:GOTO130
170 PRINT:PRINT"BINARY:  ";
180 REM **********************
190 REM ***BINARY CONVERT***
200 REM **********************
210 FOR X=15 TO 0 STEP-1
220 Y=A-2↑X
230 IF Y<0  THEN A$(X)="0"
240 IF Y>=0 THEN A$(X)="1":A=A-2↑X
250 PRINT A$(X);
260 C=C+1:IF C=4 THEN PRINT" ";:C=0
270 NEXT
280 REM ******************
290 REM ***HEX CONVERT***
300 REM ******************
310 S$="    ":K=3
320 PRINT:PRINT:PRINT"HEX:      ";
330 FOR H=15 TO 0 STEP-1
340 G=VAL(A$(H))
350 IF G=1 THEN T=T+2↑K
360 K=K-1
370 IF K<0 THEN GOSUB 400:K=3:T=0
380 NEXT
390 PRINT:FOR L=0 TO 39:PRINT"•";:NEXT:PRINT:GOTO130
400 ON T+1 GOTO 410,420,430,440,450,460,470,480,490,500,510,520,530,540,550,560
410 PRINT "0"+S$;:RETURN
420 PRINT "1"+S$;:RETURN
430 PRINT "2"+S$;:RETURN
440 PRINT "3"+S$;:RETURN
450 PRINT "4"+S$;:RETURN
460 PRINT "5"+S$;:RETURN
470 PRINT "6"+S$;:RETURN
480 PRINT "7"+S$;:RETURN
490 PRINT "8"+S$;:RETURN
500 PRINT "9"+S$;:RETURN
510 PRINT "A"+S$;:RETURN
520 PRINT "B"+S$;:RETURN
530 PRINT "C"+S$;:RETURN
540 PRINT "D"+S$;:RETURN
550 PRINT "E"+S$;:RETURN
560 PRINT "F"+S$;:RETURN
```

SHIFTED CAPITALS

When writing programs with a great deal of text, it is sometimes best to use the lower case character set. This makes for easier reading. The game Hammurabi is a good example.

To obtain the lower case mode on PET, location 59468 must be POKEd with 14 (12 returns it to graphics). Now lower case letters, plus some other graphics are available on the PET by typing the desired letter while depressing the 'shift' key. However when a lot of text is involved, constantly holding down the shift key can become rather awkward. A 'keyboard inverter' would certainly be desirable.

P.T. Spencer, a high school teacher in Agincourt, enquired about this possibility and then answered his own question with the following program which he has submitted for the Transactor.

```
10000 REM**KEYBOARD INVERTER SUBROUTINE**
10010 POKE 59468,14
10020 Z$=""
10040 GET Y$:IF Y$="" THEN 10040
10050 Y=ASC(Y$):IF Y=13 THEN 10120
10060 IF Y>64 AND Y<91 THEN Y=Y+128:GOTO 10090
10070 IF Y>192 AND Y<219 THEN Y=Y-128:GOTO 10090
10080 GOTO 10100
10090 Y$=CHR$(Y)
10100 PRINT Y$;:Z$=Z$+Y$:GOTO 10040
10120 PRINT:RETURN
```

The program only affects the alphabetic characters and prints everything else out normally. Also, since the characters are being displayed under program control the lines won't be entered into memory, only displayed! To do this program execution would have to be halted and then line numbers followed by ?" inserted at the beginning of every second line of text. (Be careful not to type more than 75 characters per 2 lines or inserting is impossible. Also do not stop program execution less than 5 lines from the bottom or the "BREAK - READY" signal will cause text on the very top line to be scrolled off the screen.) Once this insertion is finished, hitting 'RETURN' enters the characters as program text. Only one pitfall is that these PRINT statements will return the text without the visual continuity (i.e. 'PRINT' starts at extreme left) unless the closing quotation marks are inserted and followed by semi-colons. Therefore it would be more desirable to enter the text into DATA statements and concatenate using the READ command. Instead you would insert the line numbers followed by DATA" or you could have the program do it for you using the following modifications:

```
10000 REM **KEYDATA KEYBOARD INVERTER**
10010 POKE 59468,14
10020 Z$=""
10030 L=0:C=C+10:PRINT:PRINTC;"cl";"DATA";:PRINT CHR$(34);
10040 GET Y$:IF Y$="" THEN 10040
10050 Y=ASC(Y$):IF Y=13 THEN 10120
10060 IF Y>64 AND Y<91 THEN Y=Y+128:GOTO 10090
10070 IF Y>192 AND Y<219 THEN Y=Y-128:GOTO 10090
10080 GOTO 10100
10090 Y$=CHR$(Y)
10100 PRINT Y$;:Z$=Z$+Y$:L=L+1:IF L=70 THEN 10020
10110 GOTO 10040
10120 PRINT:STOP
```

Micro Magazine

Published monthly, MICRO is the only mag. devoted entirely to 6502 based systems. It covers software and hardware on PET, KIM-1, APPLE, AIM, SYM and virtually anything to do with the MOS 6502. MICRO has run out of back issues but they are offering "The Best of MICRO Volume 1", which covers issues 1 through 6. Cost is $6.00 plus $1.00 postage and handling, payable in US Funds. Subscriptions are also available; $15.00 (US) for 12 issues. Send your name and address, etc., to:

MICRO Magazine
P.O.Box 3,
SO. CHELMSFORD, MA.
01824.

Specify Best of MICRO and/or subscription and at which issue you wish your subscription to start. Highly recommended.

Transactor Volume 2

Yes it's renewal time. Although your Volume 1 subscription covers one more issue, here is the Volume 2 subscription form. The Transactor operates on a break-even basis. Therefore the cost of "The Transactor Volume 2" will be $15.00.

Recently I have received various requests regarding article subjects. If you have a subject idea you and other PET users would like to see discussed in future Transactors, please include it at the bottom of the order form.

.....

To receive Transactor Volume 2 bulletins, please return this form with your cheque for $15.00 annually renewable, to CBM 3370 Pharmacy Avenue, Agincourt, Ontario, M1W 2K4. Volume 1 back issues will be available at 10 dollars for a limited time only (while supplies last).

VOL 2

NAME.....

COMPANY (if applicable).....

ADDRESS.....

.....POSTAL CODE.....

RECEIPT REQUIRED? YES ☐ NO ☐
(Invoices cannot be issued for the $15.00 annual fee)

IDEAS & COMMENTS.....

.....

N E W P R O D U C T - - P E T P I L O T

FOR RELEASE 4-15-79

Commodore PET owners get full standard PILOT (the most popular computer-assisted instruction language) on a minimum size PET. Dave Gomberg, of the PETPILOT Project, announced not only a PETPILOT Language processor, but also an Editor suitable for preparing long programs of up to about 80,000 characters.

The new product features full BASIC in compute statements as well as two new keywords designed to make PILOT programming easier and faster. All language features of the most recent PILOT standard are implemented. In addition, the system has been designed to be easy to learn and use.

Because PET tapes can only move forward, there are limits, depending on memory size, to the distance a PETPILOT program may jump upward, but this does not create a barrier to the effective use of PETPILOT.

Only the tape drive supplied with the PET is required to run any PETPILOT program. While simple PETPILOT programs can be created on a one tape PET, authors writing long programs will need the second cassette drive offered by Commodore International of Palo Alto, CA, manufacturer of the PET.

The package offered by the PETPILOT Project contains both programs, a sample PETPILOT program, a teacher's manual, a quick reference card, and licenses to run the programs on a single PET. A tutorial course of two one hour lessons in effective use of PETPILOT is also available. The basic package costs $12, the tutorial is an extra $8. Both products can be ordered by specifing the PET serial number to be licensed to Dave Gomberg, 7 Gateview Court, San Francisco CA 94116.

TAX ONTARIO 1978

Last months listing of the Income Tax program requires a few revisions due to the fact that the Centronics 779 does not recognize PET graphics. The corrections are as follows:

```
111 INPUT"cdINSTRUCTI.....
112 PRINT"cdONTARIO IN.....
113 PRINT"cdFOLLOW YOUR.....
115 PRINT"cdFOR 'NIL' ITEMS.....
116 PRINT"cdFOR 'MULTIPLE' ITEMS.....
310 PRINT"!EXEMPTIONS.....
470 IF I 231E3 THEN PRINT"!NO TAX PAYABLE.....
2110 Y$="":PRINT"&@cl";
2139 IF (Z=157 OR Z=20) AND Y$  "" THEN Y$=LEFT$(Y$,
     LEN(Y$)-1):PRINT"clØ";:GOTO2147
2140 IF Z$="+" THEN PRINT"cd";:I=I+VAL(Y$):FORJ=1TO
     LEN(Y$):PRINT"cl";:NEX.....
2142 IF Z=13 AND I=Ø THEN PRINT"cu";
```

cd-Cursor Down, !-RVS On, &@-Shifted '&', cl-Cursor Left, Ø-Blank or Space, cu-Cursor Up.

T I S Workbooks

Total Information Services now has 5 workbooks relating to the PET. All are excellent, particularly PET Cassette on data file writing.

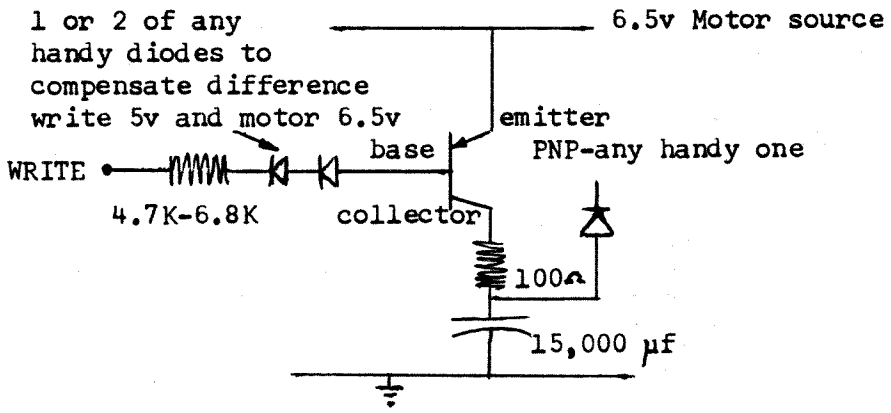
Getting Started With Your PET	-	$5.00*
PET String and Array Handling	-	$5.00*
PET Graphics	-	$6.00*
PET Cassette	-	$6.00*
PET Miscellaneous	-	$5.00*

* payable in US Funds. Write To:

Total Information Services
P.O.Box 921,
LOS ALAMOS, NM
87544

Hardware Fix

In Transactor #8, a hardware fix was illustrated to overcome the data file write problem. However, to install it required dismantling the cassette deck so the interface idea was presented. This still had one problem area: the switch. Fortunately, I've received two responses regarding the 'INTERFACE'. The first was from Jim Yost in Somerville,MA. He writes..... "In reference to Bulletin #8: The capacitor charging switch you want is a simple transistor. The basic operational rule is that whatever current flows in the base allows β times that to flow into the collector. β's typically are 100 or more nowadays. The circuit is thus:



The write pulses (lows) charge the 15,000µF cap in pulses so that it is charged at the end of a write sequence when the motor source drops. Don't use the 15µF - it would short the write pulses".

Secondly, Andrew Chiu of Toronto has designed, among other things, an interface type fix for file writing to the cassette deck. This and other devices are for sale through Rapid Electronics. Andrew has submitted a complete description of the cassette deck interface and its operation which follows.

Thank you, Andrew and Jim, and once again..... thanks to Richard Leon for launching the concept.

Andrew Chiu
39 Farmview Cr.,
WILLOWDALE, Ontario
M2J 1G5

HARDWARE FIX ADAPTER

The object of this design is to adapt Richard Lean's Hardware Fix (ref: TRANSACTOR BULLETIN #8) without dismantling your cassette and using Karl's "Interface" idea we can install a circuit between a 6 pin Molex Plug and the printed circuit board edge connector (see Fig.6).

A quick look at the table provided by Karl J. in the Transactor Bulletin #8 (table 1) indicates that when the cassette is doing a write operation the write-line is 'active' and for other operations it is 'high', therefore based on this unique active state a simple switching circuit can be controlled.

TABLE 1

PIN on PCB Cardedge	OPERATION		
	NONE	READ	WRITE
READ	LOW	ACTIVE	ACTIVE
WRITE	HIGH	HIGH	ACTIVE
SENSE	HIGH	LOW	LOW
MOTOR	OPEN	6.5V	6.5V

HOW IT WORKS. Most PET lovers are the software type, in order to understand the theory behind this circuit, let us review some characteristics of the key element, a diode.

The ideal-diode approximation strips away all but the bones of diode operation. What does a diode do? It conducts well in the forward direction and poorly in the reverse direction. Boil this down to its essence, and this is what you get: ideally, a rectifier diode acts like a perfect conductor (zero voltage) when forward-biased and like a perfect insulator (zero current) when reverse-biased as shown in Fig.3.

In circuit terms, an ideal diode acts like an automatic switch. When conventional current tries to flow in the direction of the diode arrow, the switch is closed (see Fig.3b). If conventional current tries flowing the other way, the switch is open (Fig. 3c). We cannot simplify the idea of the diode beyond this point.

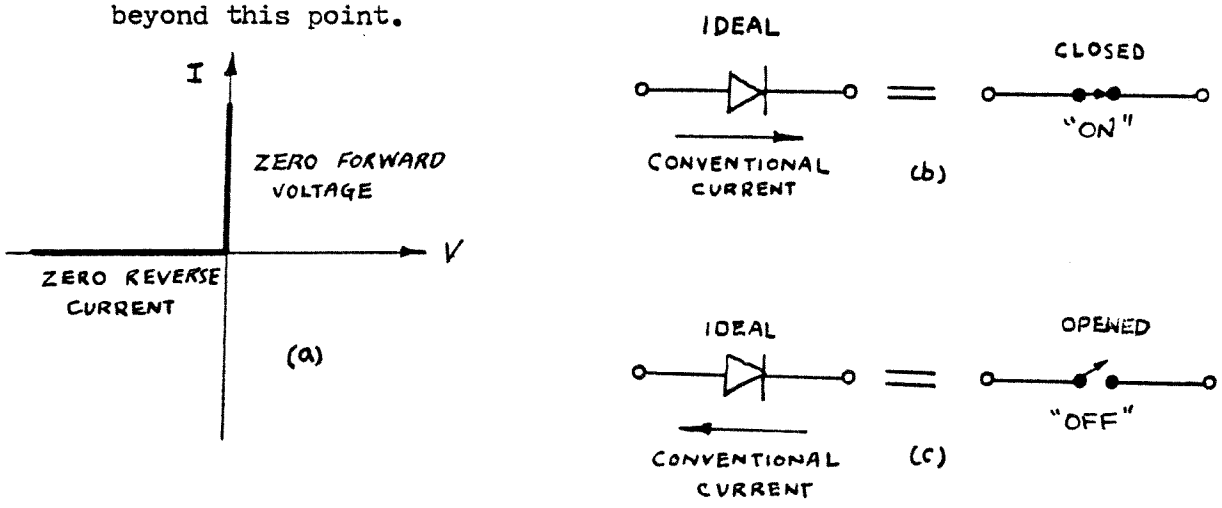


Fig 3. (a) Ideal-diode Curve. (b) Cloded-switch analogy. (c) Open-switch analogy.

Once the operation of the diode is understood, then we can look at Fig.1. It is a complete diagram of the Hardware Fix Adapter. IC.1 is connected as a negative-recovery monostable circuit. As long as the write-line is active, the circuit stays triggered and the output remains High. (approximately equal supply voltage at no load). The current flows out from pin 3 and the capacitor C3 charges through the resistor R2 and diode D2. The capacitor is acting like a temporary quick charge battery. When the block is finished, the power to the motor is turned OFF and the write-line go High. The capacitor C3 will discharge through D3 to the cassette motor. The discharged energy provided power to the motor for a small period of time. This allows a larger gap between blocks.

Fig. 4 is a simplified circuit, only the diode and C3 is shown. The imaginary switches are the approximation of IC.1 and the transistor switch which is controlled by the internal operation of your PET. Fig. 4 a & b when the motor power switch is turned ON and write-line is active, D1 and D2 are both ON, but D3 is OFF. Therefore the current has to flow into the capacitor and the motor. Fig 4 c & d when motor power switch is OFF and write-line is High, D1 and D2 is Off, but D3 is ON. Therefore the current can discharge through D3 to the motor. The ideal diode model is a helpful tool to explain the operation of the circuit.

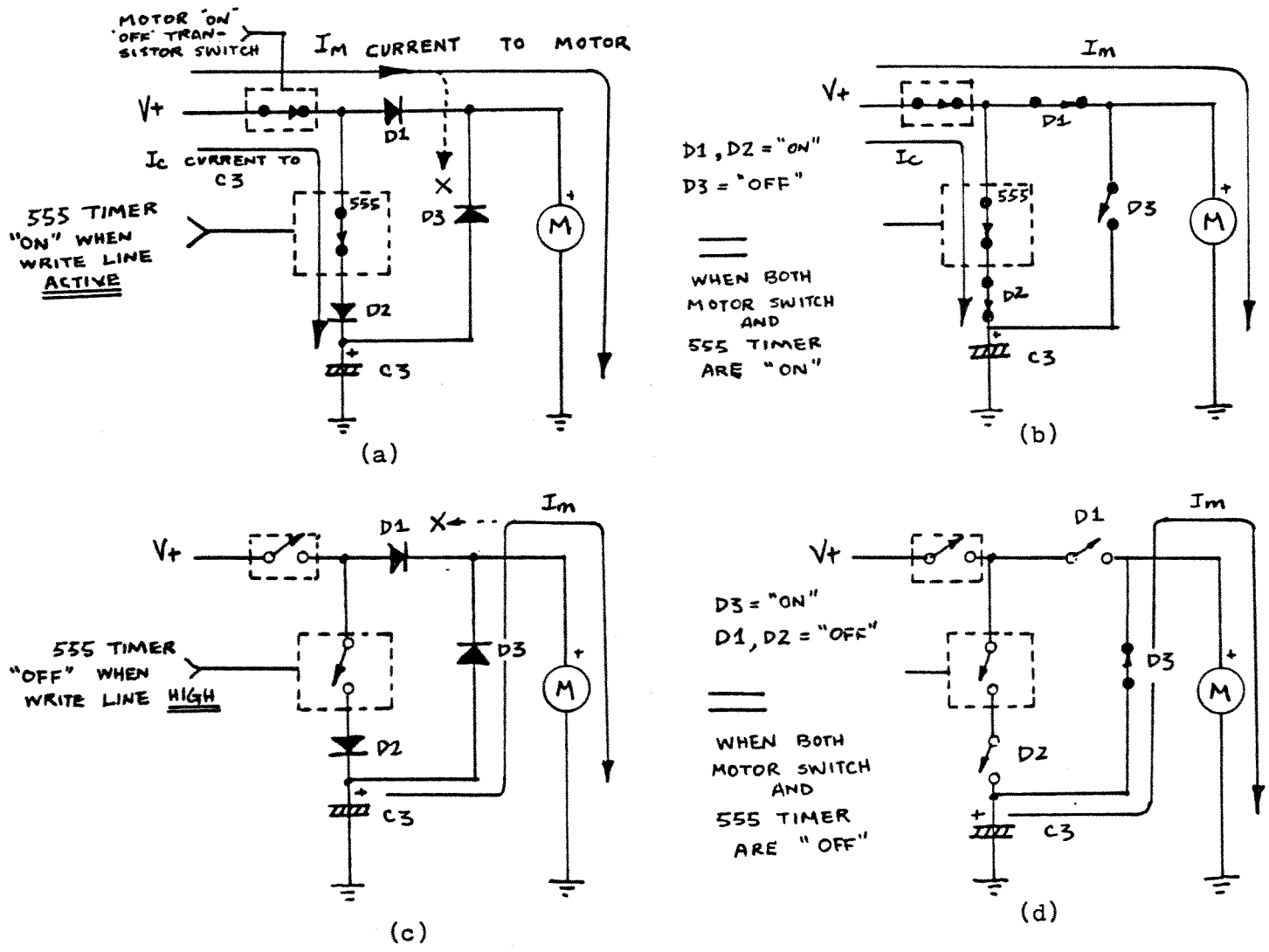


Fig. 4 Simplified circuit diagram of the Hardware Fix Adapter.

CONSTRUCTION. Wire-Wrap, point-to-point or printed-circuit techniques can be employed in the construction of the Adapter. However a printed-circut board definitely makes things easier once it has been made or obtained. I built my unit with all the components mounted on the board except the 20,000 uF capacitor. Watch the polarity of all the diodes and C3. The cathode (-Ve) end of the diode is marked with a Band or Bands. The polarity of the capacitor is marked on the label, some are colour coded. (red for positive). Transistor Q1 may be supplied in different shapes. To identify the transistor leads, please check the manufacturers data book. A component lay out diagram is shown in Fig. 2. Watch the spacing (0.156") of the card-edge connector. and the slot. The physical dimension must fit the 6 pin Molex Plug which the cassette uses.

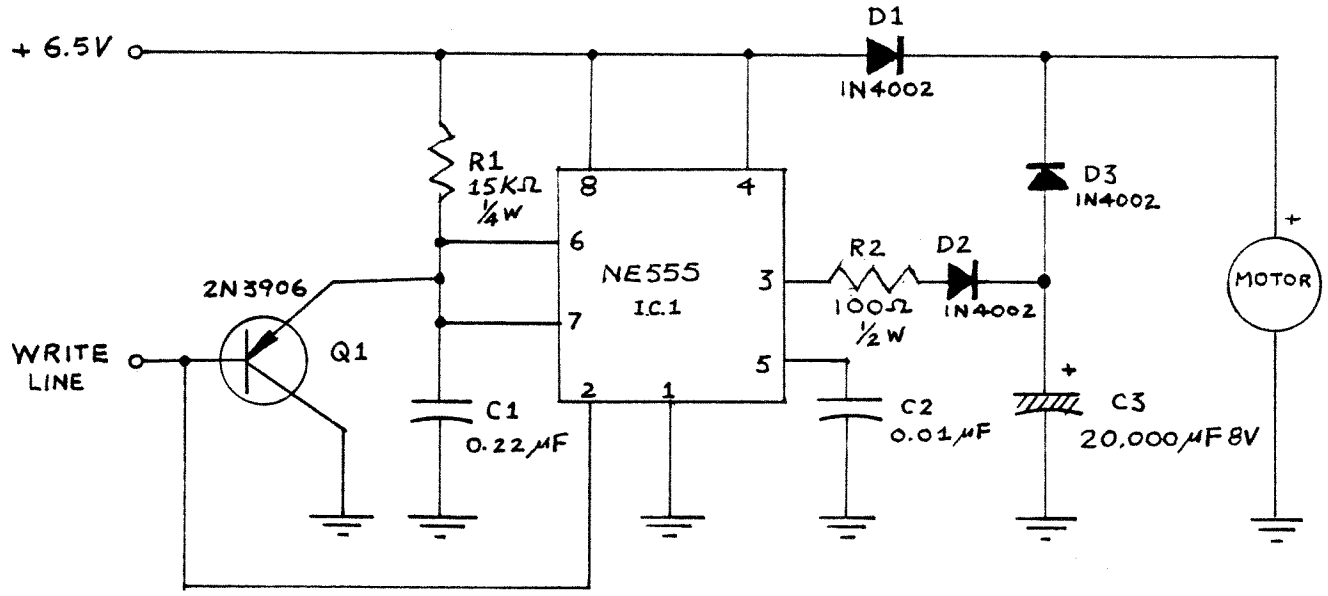


Fig 1. Circuit diagram of the Hardware Fix Adapter.

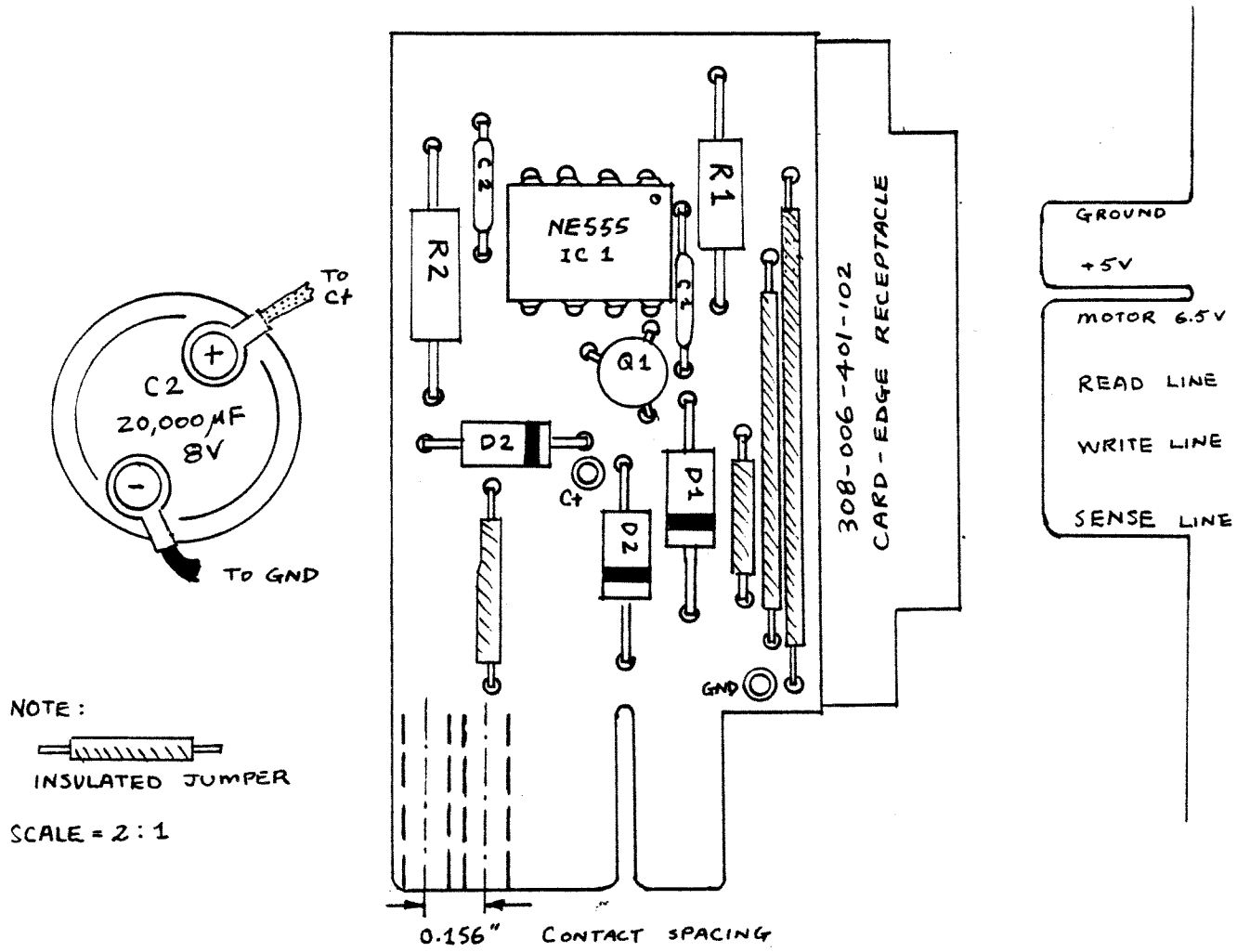


Fig 2. Parts placement, component-side view.

OPEN THE PET COMPUTER. (This is the procedure outlined in the service manual)

1. Press the rocker switch to the OFF position.
2. Remove the power cord from the wall socket to avoid possible electrical shock.
3. Remove the two screws located on each side of the unit under the lip of the cover.
4. Lift the cover slowly - a few inches. When you locate the cable leading to the cassette, remove the connector at the main board. Then left the cover all the way up and engage the supporting rod located on the left side of the cover.

INSTALLATION PROCEDURE OF THE ADAPTER. (see Fig. 5,6 & 7)

1. Plug the 6 pin Molex Plug to the Adapter's card-edge connector.
2. Plug the Adapter to the main board's card-edge connector.
3. The big 20,000 uF capacitor can be mount in front of the main board by locking cable ties and adhesive backed mounts.

FINAL TESTING. Murphy's law states that "If anything can go wrong, IT WILL". So please check the polarity of all the components again before you touch the power switch. One thing will damage your PET for sure is a short circuit between ground and +5V line or +6.5V motor control line. When you think everything is OK. Turn On your PET, and use a voltmeter to measure the voltage across C3, it should give you a very small reading (amost zero). If the voltage is right, type in the testing program. After you finished, SAVE the program, while your PET is saving the program, monitor the voltage across C3 again, it should rise slowly to approximately to 4.5V and stop there. When finished saving, the voltage will drop to zero again (0.6V). Next, put a new tape in the cassette and RUN the test program. The test program will check and print the number of drop out errors found - you should have none.

PARTS AVAILABILITY. Essential parts, including 6 pin card-edge receptacle (# K60013PCSCGD6), Universal PC Board and 20,000 uF are available from ARKON ELECTRONICS LTD., 91 Queen Street E., TORONTO, Ontario, M5C 1S1.
A assembled and tested unit for $26.00 is available from RAPID ELECTRONIC, P.O. Box 1031, Station 'B', WILLOWDALE, Ontario, M2K 2T6.

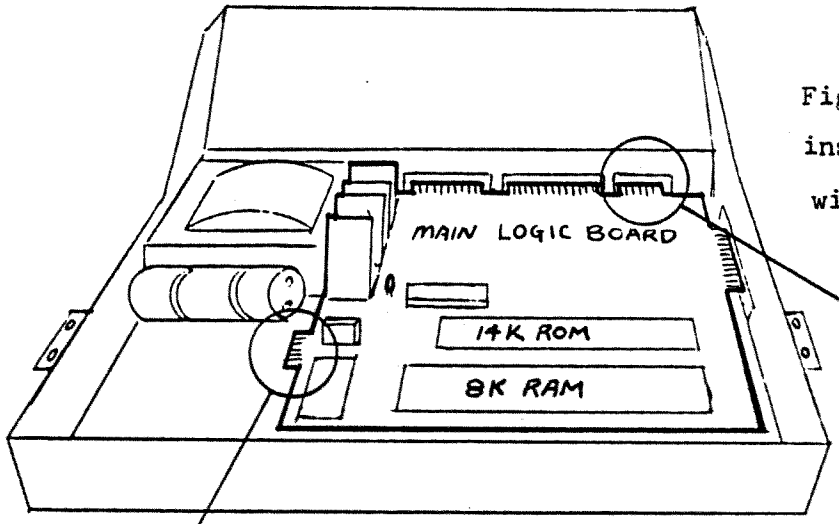


Fig 5. Simplified inside view of the PET with the cover removed.

Card-edge connector for cassette No.2 6 pin Molex Plug.

Card-edge connector for cassette No.1 6 pin Molex Plug.

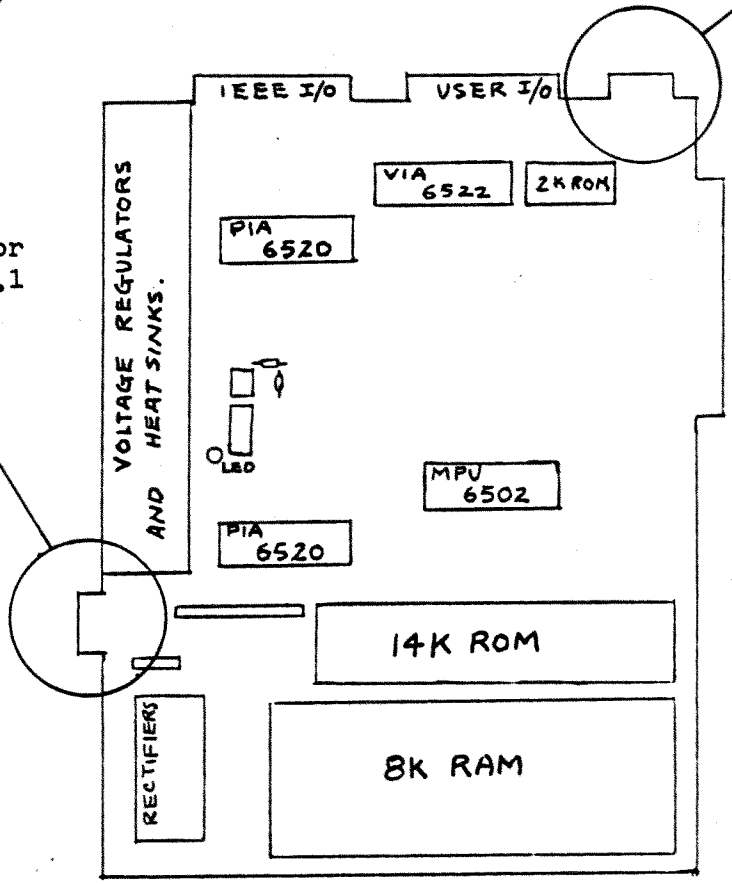
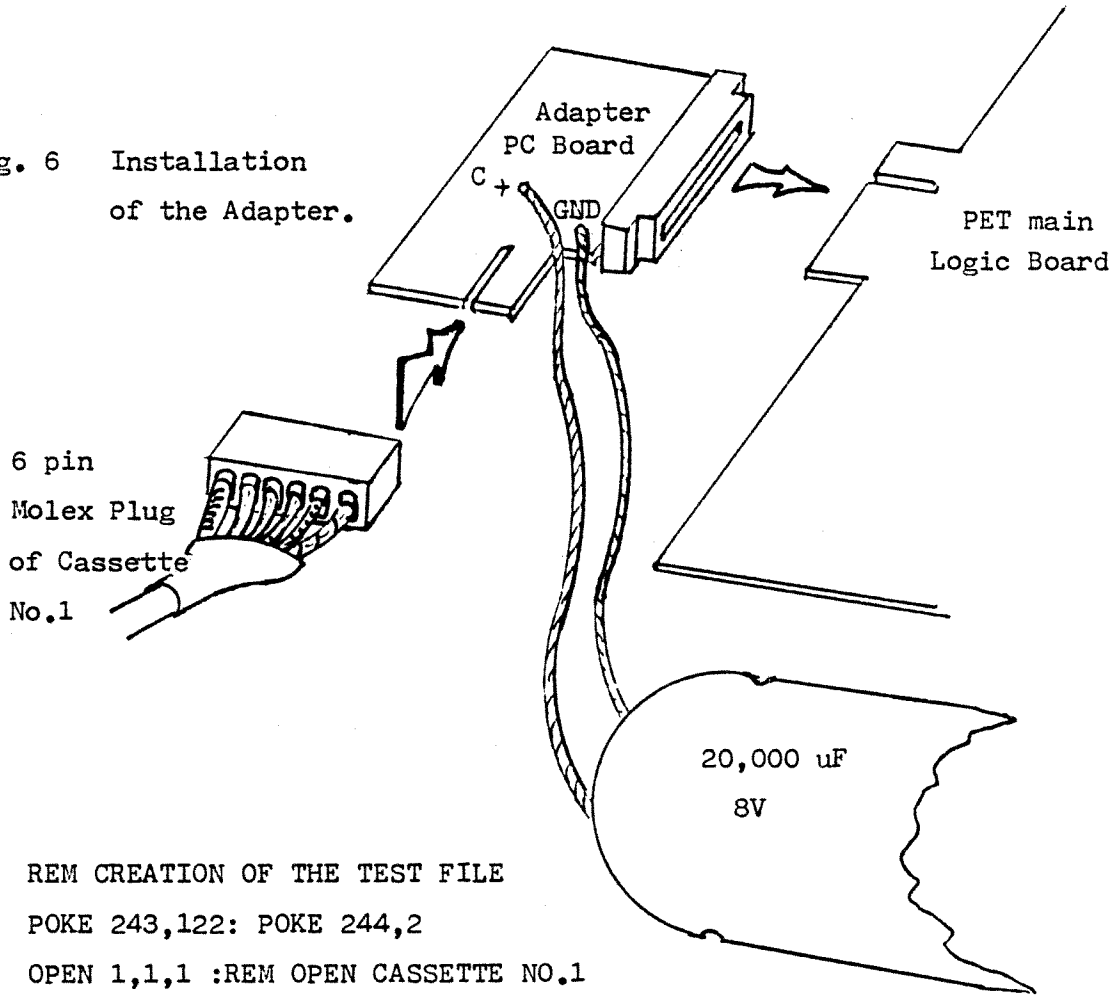


Fig 7. Simplified view of the main logic board.

Fig. 6 Installation of the Adapter.




```
1   REM CREATION OF THE TEST FILE
10  POKE 243,122: POKE 244,2
20  OPEN 1,1,1 :REM OPEN CASSETTE NO.1
30  FOR I=1 TO 300
40  PRINT I
50  PRINT#1, I
60  NEXT I
70  PRINT#1: CLOSE1
80  STOP
```

NOTE: For Cassette NO.2 the following lines have to be changed to :-

```
10 POKE 243,58:POKE 244,3
20 OPEN 1,2,1
105 OPEN 1,2
```

```
100 REM RECOVER AND TEST THE TEST FILE
105 OPEN 1
110 FOR I=1 TO 300
120 INPUT#1,A
130 IF (ST) AND 64 THEN GOTO 170
140 PRINT A;
150 IF A  B+1 THEN PRINT "ERROR": J=J+1
160 B=A: NEXT I
170 CLOSE 1: PRINT J;"ERRORS FOUND"
```